ɥ

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/847,642	05/01/2001	Luciano Lavagno	261/246	6620

23639 7590 04/30/2007
BINGHAM MCCUTCHEN LLP
Three Embarcadero Center
San Francisco, CA 94111-4067

EXAMINER	
GUILL, RUSSELL L	

ART UNIT	PAPER NUMBER
2123	

MAIL DATE	DELIVERY MODE
04/30/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)
	09/847,642	LAVAGNO ET AL.
	Examiner	Art Unit
	Russ Guill	2123

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *26 March 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,2,4-22,33,34 and 36-60* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2,4-22,33,34 and 36-60* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *27 July 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This Office Action is in response to an amendment filed March 26, 2007. Claims 23 – 32 were previously canceled. Claims 3 and 35 were canceled. No claims were added. Claims 1 – 2, 4 – 22, 33 – 34 and 36 – 60 are pending. Claims 1 – 2, 4 – 22, 33 – 34 and 36 – 60 have been examined. Claims 1 – 2, 4 – 22, 33 – 34 and 36 – 60 are rejected.

2. This Office Action is **NON-final** because of new rejections made as a result of an updated search that found new art that overcomes the Applicant's arguments and claim amendments.

3. In a previous Office Action dated April 11, 2006, the Examiner noted that the application will be forwarded to the Office of Initial Patent Examination for issuance of a corrected filing receipt, and correction of Office records to reflect the inventorship as corrected. This action is being retracted because it appears that the provisions of rule 1.48 (b) were not completely satisfied, since a processing fee does not appear to have been paid (rule 1.48 (b)(2)). However, in order to expedite the examination process, the Examiner is proceeding with examination as if the inventorship will be corrected.

4. **The Examiner would like to thank the Applicant for the well-presented amendment, which was useful in the examination process.**

Response to Remarks

5. Regarding **claims 1 - 22 and 33 – 60** rejected under 35 USC § 101:

5.1. Applicant's claim amendments have provided a tangible result for the claims, and the corresponding rejections are withdrawn. However, claims 33 – 34, 36 – 40 and 41

– 54 still appear to allow a medium that includes carrier waves and transmission media, which are non-statutory.

6. Regarding **claims 1, 9, 33, 41 and 55** rejected under 35 USC § 103:

6.1. Applicant's arguments have been fully considered, and are persuasive (pages 10 – 11); however, upon further search and consideration, new rejections are made as described below.

Claim Rejections – 35 USC § 101

7. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

8. **Claims 33 – 34 and 36 – 40 and 41 – 54** are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

8.1. Regarding **claims 33 – 34, 36 – 40 and 41 – 54**, the claims are directed to a computer program product that includes a medium useable by a processor, the medium comprising a sequence of instructions. However, the specification appears to allow the medium to include carrier waves (pages 29 – 30, paragraphs 138 – 139), which do not appear to be a process, machine, manufacture or composition of matter. Accordingly, the claims appear to be directed to a non-statutory category, and are rejected.

8.2. Regarding **claims 33 – 34, 36 – 40 and 41 – 54**, the claims are directed to a computer program product that includes a medium useable by a processor, the medium comprising a sequence of instructions. However, the specification appears to allow the medium to include transmission media such as copper wires, coaxial cable

and fiber optics (pages 29 – 30, paragraphs 138 – 139). A computer program propagating in a transmission media is not a physical element. Such claimed computer programs do not define any structural and functional interrelationship between the computer program and other elements of a computer which permit the computer program's functionality to be realized. Accordingly, transmission media appear to be a non-statutory category, and the claims are rejected.

Claim Rejections – 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

11. **Claims 1 – 2, 4 - 13, 15 – 19, 21 - 22, 33 – 34, 36 - 45, 47 – 51, 53 – 55, 57 – 58 and 60** are rejected under 35 U.S.C. 103(a) as being unpatentable over Passerone (Claudio Passerone et al.; "Fast hardware/software co-simulation for virtual prototyping and trade-off analysis", 1997, Proceedings of Design Automation Conference 1997) in view of Hellestrand (U.S. Patent Number 6,230,114), further in view of Zivojnovic (Vojin Zivojnovic, "Compiled HW/SW Co-simulation", 1996, art supplied by the Applicant on the Information Disclosure Statement dated September 10, 2001, element AP).

11.1. Regarding **claims 1 and 33**:

11.2. Passerone appears to teach:

11.2.1. Describing a system design as a network of logical entities (***page 3, section 2.2, first paragraph, the sentence, "The formal specification of the system to be modeled is first translated by POLIS into a network of CFSMs, and then synthesized as timing-annotated C code as described above"***);

11.2.2. Selecting at least one of the logical entities for a software implementation (***page 3, section 2.2, first paragraph, the sentence, "The formal specification of the system to be modeled is first translated by POLIS into a network of CFSMs, and then synthesized as timing-annotated C code as described above"***);

11.2.3. Implementing a source software program from the logical entities selected for the software implementation (***page 3, section 2.2, first paragraph, the sentence, "The formal specification of the system to be modeled is first translated by POLIS into a network of CFSMs, and then synthesized as timing-annotated C code as described above"***);

11.2.4. Compiling the software program to generate an optimized assembler code representation of the software program (***page 3, section 2.2, first paragraph, the sentence, "The formal specification of the system to be modeled is first translated by POLIS into a network of CFSMs, and then synthesized as timing-annotated C code as described above"; it would have been obvious that C code is compiled in order to run a simulation, and that a compiler generates optimized assembler code***);

11.2.5. Generating a hardware/software co-simulation model using the simulation model (***page 3, section 2.2, first paragraph, the sentences, "The formal specification of the system to be modeled is first translated by POLIS into a network of CFSMs, and then synthesized as timing-annotated C code as described above" and "The C code is used in Ptolemy as a model for both hardware and software components"; and page 1, right-side column, third paragraph that starts with, "It is based on . . ."***);

11.2.6. storing at least the hardware/software co-simulation model (***page 3, section 2.2, first paragraph, the sentences, "The formal specification of the system to be modeled is first translated by POLIS into a network of CFSMs, and then synthesized as timing-annotated C code as described above" and "The C code is used in Ptolemy as a model for both hardware and software components"; and page 1, right-side column, third paragraph that starts with, "It is based on . . ."; it would have been obvious to the ordinary artisan to save the model in order to be able to run the model multiple times***).

11.3. Passerone does not specifically teach:

11.3.1. Performing a performance analysis using the assembler code;

11.3.2. Generating an assembler-level C software simulation model using the assembler code;

11.4. Hellestrand appears to teach:

11.4.1. Performing a performance analysis using the assembler code (***figure 3A, elements 311, 313; and column 26, lines 6 – 18***);

11.4.2. Generating a software simulation model using the assembler code (***figure 3A, elements 311, 313, 315, 317, 319, 331, 333; and column 32, lines 14 – 36***);

11.5. Zivojnovic appears to teach:

11.5.1. Generating an assembler-level C software simulation model using the assembler code (***page 692, section IV Compiled Simulation of Programmable Architectures; and page 693, figure 1 b)***);

11.6. The motivation to use the art of Hellestrand with the art of Passerone would have been the benefit recited in Hellestrand that there is an important advantage to the system – a linear block can be as short as a single instruction, and the user has the option of so analyzing the code to get instruction-by-instruction timing (***column 25, lines 27 – 33***).

11.7. The motivation to use the art of Zivojnovic with the art of Passerone would have been the benefit recited in Zivojnovic that the method offers up to three orders of magnitude faster simulation (**page 690, Abstract**).

11.8. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Hellestrand and Zivojnovic and Passerone to produce the claimed invention.

11.9. Regarding **claims 2 and 34**:

11.10. Passerone appears to teach:

11.10.1. the compiling step further comprises incorporating a description of the target machine (***page 1, right-side column, third paragraph that starts with, "It is based on using . . ."***);

11.11. Regarding **claims 4 and 36**:

11.12. Passerone appears to teach:

11.12.1. selecting at least one of the logical entities for a hardware implementation, and synthesizing a software model of the hardware implementation from the selected logical entities, wherein the hardware/software co-simulation model is generated using the software model of the hardware implementation (***page 3, section 2.2, first paragraph, the sentences, "The formal specification of the system to be modeled is first translated by POLIS into a network of CFSMs, and then synthesized as timing-annotated C code as described above" and "The C code is used in Ptolemy as a model for both hardware and software components"; and page 1, right-side column, third paragraph that starts with, "It is based on . . ."***);

11.13. Regarding **claims 5 and 37**:

11.14. Passerone does not specifically teach:

11.14.1. the performance analysis measures an execution time of an element of the assembler code.

11.15. Hellestrand appears to teach:

11.15.1. the performance analysis measures an execution time of an element of the assembler code (***figure 3A, elements 311, 313; and column 26, lines 6 – 18***).

11.16. Regarding **claims 6 and 38**:

11.17. Passerone does not specifically teach:

11.17.1. the software program is compiled using the same compiler used to compile a production executable.

11.18. Hellestrand appears to teach:

11.18.1. the software program is compiled using the same compiler used to compile a production executable (***figure 3A, elements 331, 333***).

11.19. Regarding **claims 7 and 39**:

11.20. Passerone appears to teach:

11.20.1. performing the performance analysis comprises annotating the code with performance information (***page 3, section 2.2, first paragraph, the sentences, "The formal specification of the system to be modeled is first translated by POLIS into a network of CFSMs, and then synthesized as timing-annotated C code as described above"***).

11.21. Passerone does not specifically teach:

11.21.1. performing the performance analysis comprises annotating the ***assembler*** code with performance information.

11.22. Hellestrand appears to teach:

11.22.1. assembler code (***figure 3A, element 311; it would have been obvious to annotate assembler code because it was old and well known to annotate code with performance information; for example, Verilog HDL includes syntax to annotate code with timing information***);

11.23. Regarding **claims 8 and 40**:

11.24. Passerone appears to teach:

11.24.1. the performance information is timing information (***page 3, section 2.2, first paragraph, the sentences, "The formal specification of the system to be modeled is first translated by POLIS into a network of CFSMs, and then synthesized as timing-annotated C code as described above"***).

11.25. Regarding **claims 9 and 41**:

11.26. Passerone appears to teach:

11.26.1. obtaining a software assembly code module from a source code module (***page 3, section 2.2, first paragraph, the sentence, "The formal specification of the system to be modeled is first translated by POLIS into a network of CFSMs, and then synthesized as timing-annotated C code as described above"***);

11.26.2. translating the software code module into a simulation model (***page 3, section 2.2, first paragraph, the sentences, "The formal specification of the system to be modeled is first translated by POLIS into a network of CFSMs, and then synthesized as timing-annotated C code as described above" and "The C code is used in Ptolemy as a model for both hardware and software components"; and page 1, right-side column, third paragraph that starts with, "It is based on . . ."***);

11.26.3. annotating the software simulation model with performance information (***page 3, section 2.2, first paragraph, the sentence, "The formal specification of the system to be modeled is first translated by POLIS into a network of CFSMs, and then synthesized as timing-annotated C code as described above"; further, it was old and well known in the art to annotate code with performance information; for example, Verilog HDL includes syntax to annotate code with timing information***);

11.26.4. storing at least the software simulation model (***page 3, section 2.2, first paragraph, the sentences, "The formal specification of the system to be modeled is first translated by POLIS into a network of CFSMs, and then synthesized as timing-annotated C code as described above" and "The C code is used in Ptolemy as a model for both hardware and software components"; and page 1, right-side column, third paragraph that starts with, "It is based on . . ."; it would have been obvious to the ordinary artisan to save the model in order to be able to run the model multiple times***);

11.27. Passerone does not specifically teach (the missing parts are indicated in ***bold, italic, underline***):

11.27.1. obtaining a software assembly code module from a source code module;

11.27.2. translating the assembly code module into a software simulation model;

11.27.3. wherein the software simulation model is an assembler-level software simulation model, expressed in a high-level programming language.

11.28. Hellestrand appears to teach:

11.28.1. obtaining a software **assembly** code module from a source code module (***figure 3A, elements 319, 331, 333***);

11.29. Zivojnovic appears to teach:

11.29.1. translating the assembly code module into a software simulation model (***page 692, section IV Compiled Simulation of Programmable Architectures; and page 693, figure 1 b)***);

11.29.2. wherein the software simulation model is an assembler-level software simulation model, expressed in a high-level programming language (***page 692, section IV Compiled Simulation of Programmable Architectures; and page 693, figure 1 b)***).

11.30. Regarding **claims 10 and 42**:

11.31. Passerone does not specifically teach:

11.31.1. providing a software assembly code module comprises compiling software source code to assembly.

11.32. Hellestrand appears to teach:

11.32.1. providing a software assembly code module comprises compiling software source code to assembly (***figure 3A, elements 309, 311***).

11.33. Regarding **claims 11 and 43**:

11.34. Passerone appears to teach:

11.34.1. the software code module is compiled using a compiler adapted to create code that will execute on a first machine architecture (***page 3, section 2.2, first paragraph, the sentence, "The formal specification of the system to be modeled is first translated by POLIS into a network of CFSMs, and then synthesized as timing-annotated C code as described above"; it would have been obvious that the C code was compiled***);

11.35. Passerone does not specifically teach:

11.35.1. assembly code

11.36. Hellestrand appears to teach:

11.36.1. assembly code (***figure 3A, element 311***).

11.37. Regarding **claims 12 and 44**:

11.38. Passerone appears to teach:

11.38.1. the performance information is associated with the first machine architecture (***page 3, section 2.2, first paragraph, the sentence, "The formal specification of the system to be modeled is first translated by POLIS into a network of CFSMs, and then synthesized as timing-annotated C code as described above"***);

11.39. Regarding **claims 13 and 45**:

11.40. Passerone appears to teach:

11.40.1. the simulation model is compiled to execute on a second machine architecture, the second machine architecture being different from the first machine architecture (***page 3, section 2.2, first paragraph, the sentence, "The formal specification of the system to be modeled is first translated by POLIS into a network of CFSMs, and then synthesized as timing-annotated C code as described above"***);

11.41. Regarding **claims 15 and 47**:

11.42. Passerone does not specifically teach:

11.42.1. the high-level programming language comprises a C code programming language;

11.43. Zivojnovic appears to teach:

11.43.1. the high-level programming language comprises a C code programming language (***page 693, figure 1, section b***);

11.44. Regarding **claims 16 and 48**:

11.45. Passerone appears to teach:

11.45.1. the translation step further comprises gathering information from the source code module from which the assembly code module was obtained (***page 3, section 2.2, first paragraph, the sentence, "The formal specification of the system to be modeled is first translated by POLIS into a network of CFSMs, and then synthesized as timing-annotated C code as described above"; it would have been obvious that in order to annotate code with timing information that source code module provided information***);

11.46. Regarding **claims 17 and 49**:

11.47. Passerone does not appear to teach:

11.47.1. information gathered comprises high-level hints about the software assembly code module.

11.48. Hellestrand appears to teach:

11.48.1. information gathered comprises high-level hints about the software assembly code module (***figure 3A, elements 313, 315***).

11.49. Regarding **claims 18 and 50**:

11.50. Passerone does not appear to teach:

11.50.1. the performance information comprises estimated performance information.

11.51. Hellestrand appears to teach:

11.51.1. the performance information comprises estimated performance information (***figure 3A, elements 313, 315***).

11.52. Regarding **claims 19 and 51**:

11.53. Passerone does not appear to teach:

11.53.1. the performance information is statically estimated.

11.54. Hellestrand appears to teach:

11.54.1. the performance information is statically estimated (***figure 3A, elements 313, 315***).

11.55. Regarding **claims 21 and 53**:

11.56. Passerone appears to teach:

11.56.1. compiling the simulation model to a simulator host program (***page 3, section 2.2, first paragraph, the sentences, "The formal specification of the system to be modeled is first translated by POLIS into a network of CFSMs, and then synthesized as timing-annotated C code as described above"; and page 1, right-side column, third paragraph that starts with, "It is based on . . ."***);

11.56.2. executing the simulator host program on a simulator to allow performance measurements to be taken (***page 3, section 2.2, first paragraph, the sentences, "The formal specification of the system to be modeled is first translated by POLIS into a network of CFSMs, and then synthesized as timing-annotated C code as described above"; and page 1, right-side column, third paragraph that starts with, "It is based on . . ."***);

11.57. Regarding **claims 22 and 54**:

11.58. Passerone appears to teach:

11.58.1. linking an already annotated module with the simulation model (***page 3, section 2.2, first paragraph, the sentences, "The formal specification of the***

system to be modeled is first translated by POLIS into a network of CFSMs, and then synthesized as timing-annotated C code as described above"; and page 1, right-side column, third paragraph that starts with, "It is based on . . .");

11.59. Regarding **claim 55**:

11.60. Passerone appears to teach:

11.60.1. associating performance information with an element of the software module (***page 3, section 2.2, first paragraph, the sentence, "The formal specification of the system to be modeled is first translated by POLIS into a network of CFSMs, and then synthesized as timing-annotated C code as described above"***);

11.60.2.

11.61. Passerone does not specifically teach (the missing parts are indicated in ***bold, italic, underline***):

11.61.1. ***receiving the assembly language software module***;

11.61.2. ***parsing the assembly language software module into a data structure, the data structure comprising one or more nodes, each of the one or more nodes being mapped to a period of time using a mapping definition, each of the one or more nodes containing an element of the assembly language software module***;

11.61.3. ***processing the data structure to refine the accuracy of the simulation model by generating an assembler-level software simulation model using the assembly language software module, wherein the assembler-level software simulation model is expressed in a high-level programming language***;

11.61.4. associating performance information with an element of the ***assembly language*** software module;

11.61.5. ***storing and outputting the simulation model***;

11.62. Hellestrand appears to teach:

11.62.1. receiving the assembly language software module (***figure 3A, element 311***);

11.62.2. parsing the assembly language software module into a data structure, the data structure comprising one or more nodes, each of the one or more nodes being mapped to a period of time using a mapping definition, each of the one or more nodes containing an element of the assembly language software module (***figure 3A, elements 313, 315***);

11.62.3. processing the data structure to refine the accuracy of the simulation model (***figure 3A, elements 313, 315***);

11.62.4. associating performance information with an element of the assembly language software module (***figure 3A, elements 311, 313; and column 26, lines 6 – 18***);

11.62.5. storing and outputting the simulation model (***figure 3A, elements 331, 333***);

11.63. Zivojnovic appears to teach:

11.63.1. generating an assembler-level software simulation model using the assembly language software module, wherein the assembler-level software simulation model is expressed in a high-level programming language (***page 692, section IV Compiled Simulation of Programmable Architectures; and page 693, figure 1 b***);

11.64. Regarding **claim 57**:

11.65. Passerone does not specifically to teach:

11.65.1. the performance information comprises an execution delay value for the element of the assembly language software module.

11.66. Hellestrand appears to teach:

11.66.1. the performance information comprises an execution delay value for the element of the assembly language software module (***figure 3A, elements 311, 313, 315, 317, 303, 319***).

11.67. Regarding **claim 58**:

11.68. Passerone does not specifically to teach:

11.68.1. the performance information is a statically computed value.

11.69. Hellestrand appears to teach:

11.69.1. the performance information is a statically computed value (***figure 3A, elements 311, 313, 315, 317, 303, 319***).

11.70. Regarding **claim 60**:

11.71. Passerone does not specifically to teach:

11.71.1. processing the data structure comprises replicating the behavior of the assembly language software model in the simulation model.

11.72. Hellestrand appears to teach:

11.72.1. processing the data structure comprises replicating the behavior of the assembly language software model in the simulation model (***figure 3A, elements 311, 313, 315, 317, 303, 319***).

12. Claims 14 and 46 are rejected under 35 U.S.C. 103(a) as being unpatentable over Passerone as modified by Hellestrand and Zivojnovic as applied to claims **1 – 2, 4 - 13, 15 – 19, 21 - 22, 33 – 34, 36 - 45, 47 – 51, 53 – 55, 57 – 58 and 60** above, further in view of Hartoog (Hartoog, Mark R.; Rowson, James A.; Reddy, Prakash D.; Desai, Soumya; Dunlop, Douglas D.; Harcourt, Edwin A.; Khullar, Neeti; "Generation of Software Tools from Processor Descriptions for Hardware/Software Codesign", Proceedings of the 34th Design Automation Conference, June 9 – 13 1997).

12.1. Passerone as modified by Hellestrand and Zivojnovic teaches a method for preparing software for a performance estimation as applied to claims **1 – 2, 4 - 13, 15 – 19, 21 - 22, 33 – 34, 36 - 45, 47 – 51, 53 – 55, 57 – 58 and 60** above.

12.2. Regarding **claims 14 and 46**:

12.3. Passerone as modified by Hellestrand and Zivojnovic does not specifically teach:

12.3.1. disassembling software binary code to assembly code.

12.4. Hartoog appears to teach:

12.4.1. disassembling software binary code to assembly code (***page 305, section 5***).

12.5. The motivation to use the art of Hartoog with the art of Passerone as modified by Hellestrand and Zivojnovic would have been the benefit recited in Hartoog that, using a declarative description of an instruction set makes it possible to automatically generate several useful tools such as an Instruction Set Simulator and a Compiled Instruction Set Simulator (***page 304, section 3. Tools, first paragraph***), which would have been recognized as important benefit to save time by the ordinary artisan.

12.6. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Hartoog with the art of Passerone as modified by Hellestrand and Zivojnovic to produce the claimed invention.

13. Claims 20, 52 and 59 are rejected under 35 U.S.C. 103(a) as being unpatentable over Passerone as modified by Hellestrand and Zivojnovic as applied to claims **1 – 2, 4 - 13, 15 – 19, 21 - 22, 33 – 34, 36 - 45, 47 – 51, 53 – 55, 57 – 58 and 60** above, further in view of

Suzuki (Suzuki, Kei; Sangiovanni-Vincentelli, Alberto; "Efficient Software Performance Estimation Methods for Hardware/Software Codesign", 1996, Proceedings of the 33rd annual conference on Design Automation).

13.1. Passerone as modified by Hellestrand and Zivojnovic teaches a method for preparing software for a performance estimation as applied to claims **1 – 2, 4 - 13, 15 – 19, 21 - 22, 33 – 34, 36 - 45, 47 – 51, 53 – 55, 57 – 58 and 60** above.

13.2. Regarding **claims 20 and 52**:

13.3. Passerone as modified by Hellestrand and Zivojnovic does not specifically teach:

13.3.1. performance information is computed dynamically at run-time, using a formula provided during the annotating step.

13.4. Suzuki appears to teach:

13.4.1. performance information is computed dynamically at run-time, using a formula provided during the annotating step (***page 5, figure 4, run-time formula to calculate C_t.max_time***).

13.4.1.1. Regarding (***page 5, figure 4, run-time calculation of C_t.max_time***); it would have been obvious to use the dynamic run-time formula as the annotation.

13.5. Regarding **claim 59**:

13.6. Passerone as modified by Hellestrand and Zivojnovic does not specifically teach:

13.6.1. performance information is a formula for dynamically computing a value.

13.7. Suzuki appears to teach:

13.7.1. performance information is a formula for dynamically computing a value (***page 5, figure 4, run-time formula to calculate C_i.max_time***).

13.8. The motivation to use the art of Suzuki with the art of Passerone as modified by Hellestrand and Zivojnovic would have been the benefit recited in Suzuki that, two methods are presented for accurate and fast estimation of software performance in embedded real-time reactive systems designed with the POLIS system (***page 1, section 1. Introduction, second paragraph that starts with, "In this paper, we . . ."***), which would have been recognized as important benefit to save time by the ordinary artisan.

13.9. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Suzuki with the art of Passerone as modified by Hellestrand and Zivojnovic to produce the claimed invention.

14. Examiner's Note: Examiner has cited particular columns and line numbers in the references applied to the claims above for the convenience of the applicant. Although the specified citations are representative of the teachings of the art and are applied to specific limitations within the individual claim, other passages and figures may apply as well. It is respectfully requested from the Applicant in preparing responses, to fully consider the references in their entirety as potentially teaching all or part of the claimed invention, as well as the context of the passage as taught by the prior art or disclosed by the Examiner. The entire reference is considered to provide disclosure relating to the claimed invention.

Conclusion

15. The art made of record and not relied upon is considered pertinent to the applicant's disclosure:

15.1. Jwahar R. Bammi et al.; "Software Performance Estimation Strategies in a System-Level Design Tool", May 3, 2000, CODES 2000; while not prior art, the reference provides an explanation of the synthesis of the instant application and points to supporting prior art.

16. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Russ Guill whose telephone number is 571-272-7955. The examiner can normally be reached on Monday – Friday 10:00 AM – 6:30 PM.

17. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Paul Rodriguez can be reached on 571-272-3753. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300. Any inquiry of a general nature or relating to the status of this application should be directed to the TC2100 Group Receptionist: 571-272-2100.

18. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Russ Guill

Examiner

Art Unit 2123

RG

ZOILA CABRERA
PRIMARY EXAMINER
TECHNOLOGY CENTER 2100

4/23/07